Combining Constraint Types From Public Data in Aerial Image Segmentation

Jacobsen, Thomas Stig ; Jensen, Jacob Jon ; Jensen, Daniel Rune ; Samuelsen, Niels Nørgaard

Published in:
Twelfth Scandinavian Conference on Artificial Intelligence

DOI (link to publication from Publisher):
10.3233/978-1-61499-330-8-125

Publication date:
2013

Document Version
Early version, also known as pre-print

Twelfth Scandinavian Conference on Artificial Intelligence
M. Jaeger et al. (Eds.)
IOS Press, 2013

doi:10.3233/978-1-61499-330-8-125

Combining Constraint Types From Public Data in Aerial Image Segmentation

Thomas Stig JACOBSEN [a] Jacob Jon JENSEN [a], Daniel Rune JENSEN [a]
Niels Nørgaard SAMUELSEN [a]
[a] *Department of Computer Science, Aalborg University*

Abstract. We introduce a method for image segmentation that constraints the clustering with map and point data. The method is showcased by applying the spectral clustering algorithm on aerial images for building detection with constraints built from a height map and address point data. We automatically detect the number of clusters using the elongated K-means algorithm instead of using the standard spectral clustering approach of a predefined number of clusters. The results are refined by filtering out noise with a binary morphological operator. Point data is used for semi-supervised labelling of building clusters. Our evaluation show that the combination of constraints have a positive impact on the clustering quality achieved. Finally we argue how the presented constraint types may be used in other applications.

Keywords. image segmentation, spectral clustering, LIDAR, constraint

1. Introduction

The field of satellite and aerial image segmentation have accomplished much in recent years and a large variety of methods have been introduced for various applications. Road extraction is an area that has received a lot of attention as it has several daily uses such as finding quickest paths, and can be done effectively using GPS tracing data. Other methods have been developed for more specific applications, such as building detection which has several practical uses in for example city and disaster management [1].

With the large amount of geographically related data made available to the public in recent years, we believe that new and interesting methods of image segmentation are made possible. The combination of several distinct datasets could result in a better segmentation compared to using any specific dataset individually. Examples of two different types of datasets that have been made available to the public are LIDAR height data of Denmark [2] and address coordinates covering Denmark [3]. Other datasets can be found using repositories of public data sources such as [4]. The referenced LIDAR dataset can help in identifying buildings based on differences in height on the map. In addition, the LIDAR data can be combined with data from the Building- and Residence Register of Denmark (BBR) which contain the longitude and latitude coordinates of all registered addresses in Denmark for detecting buildings. We suggest that these datasets can be used to create constraints for clustering on aerial images. The LIDAR dataset takes the form of a map and the BBR data are point data. It is interesting to examine how these two types can be combined and utilised in the clustering.

The purpose of this paper is to present a general method for incorporating two constraint types in a manner that is not specific to an application of image segmentation. To show the usefulness of this method, we choose to showcase it by showing its viability in building detection. This case may in turn serve as inspiration for other applications of the method.

In this paper, we adopt the Constrained Spectral Clustering (CSC) technique from [5] and show how it can be utilised for building detection on aerial images. Our main contribution is showing how to incorporate side information specifically in the form of map and point data that can be related geographically to the images for improved clustering. The improvement of the clustering is archived by incorporating the datasets as both soft and hard constraints in the spectral clustering algorithm. Semi-supervised labelling is performed as a post-processing step with the BBR data which provide points that must be labelled as buildings. Finally, we introduce a binary morphology operator to filter out noise in the clustering.

2. Related Work

CSC is a popular clustering technique due to the flexibility of spectral clustering, and the domain adaptability when adding constraints. The strengths of spectral clustering in its pure form has been determined and verified in many applications [6]. These strengths can be explained by the similarities between spectral clustering, the graph cut problem and the graph random walk problem.

The added potential of constraints in combination with spectral clustering has also shown obvious advantages in several papers [7,8,9], using distinct techniques for applying constraints to the affinity matrix, Laplacian matrix or the eigenspace.

The incorporation of large amounts of constraints in the converted form of pairwise constraints have been illustrated more recently as a framework in the paper [5]. Their method consists of encoding the constraints in a constraint matrix alongside the affinity matrix. This work is especially relevant when considering large datasets for constraints, such as the LIDAR height data.

3. Background

We will in this section present the datasets used for the demonstration of our method and further describe the CSC algorithm.

3.1. Datasets

The LIDAR datasets [2] is a point cloud of height data that have been preprocessed into a surface grid map with a resolution of 1.6 meters. There exists two versions of this data. The first is the surface map, where both terrain and structures are present. The second is further processed into a terrain map. In the terrain map, objects are removed by smoothing any area where a rise in elevation exceeds a certain threshold determined by the BBR. By computing the difference between the two maps we construct a terrain-normalised object map only containing the height data of objects, as we are not interested in terrain height for building detection. We have also converted this map into a greyscale image

to easier interpret the height data alongside aerial images. For presentation purposes, the greyscale luminosity is scaled such that a height difference of 0.25 meters equals a difference of 1 in the range $(0,255)$ (from black to white). From here on whenever we refer to LIDAR map data, we refer to the greyscale images containing the terrain-normalised object map data.

The BBR [3] has a database with longitude and latitude of every registered address in Denmark. The data of these addresses is not perfect since not every registered address is associated with a building. Furthermore, when a building is present, the coordinates does not always correspond to the top of a building. This error margin has to be addressed when using this data as building indicators.

3.2. Spectral Clustering

Spectral clustering is a clustering technique that has recently become a very popular clustering method for non-trivial clustering problems, especially in the domain of image segmentation. We will work with the framework presented in [5] for applying constraints to spectral clustering. The CSC approach can be divided into three parts. First it generates candidates by solving a generalised eigenvalue system. The feasible set of candidates are then found by removing eigenvectors associated with non-negative eigenvalues and the rest is normalised. Finally, these feasible eigenvectors are sorted with respect to minimising the cost of the cut. The remaining eigenvectors are gathered in a matrix in which the columns define a new space and the rows are the points used for clustering. An advantage to this approach is that inconsistent constraints are managed by choosing the optimal solution among the feasible eigenvectors that satisfies enough constraints based on a lower bound β.

Input : Affinity matrix A, constrained matrix Q, belief parameter β, number of clusters K
Output: The cluster assignment indicator $\mathbf{u}^*$

$vol \leftarrow \Sigma_{i=1}^{n}\Sigma_{j=1}^{n}A_{ij}, \quad D \leftarrow \text{diag}(\Sigma_{j=1}^{n}A_{ij})$
$\bar{L} \leftarrow I - D^{-1/2}AD^{-1/2}, \quad \bar{Q} \leftarrow D^{-1/2}QD^{-1/2}$
$\lambda_{max} \leftarrow$ the largest eigenvalue of $\bar{Q}$
if $\beta \geq \lambda_{K-1}(\bar{Q}) \cdot vol$ **then**
  return $\mathbf{u}^* \leftarrow \emptyset$
else
  Solve the generalised eigenvalue problem $\bar{L}\mathbf{v} = \lambda(\bar{Q} - \frac{\beta}{vol}I)\mathbf{v}$;
  Remove eigenvectors associated with non-positive eigenvalues and normalise the rest by $\mathbf{v} \leftarrow \frac{\mathbf{v}}{||\mathbf{v}||}\sqrt{vol}$;
  $V^* \leftarrow argmin_{V \in \mathbb{R}^{N\times(K-1)}} \text{trace}(V^T\bar{L}V)$;
  return $\mathbf{u}^* \leftarrow kmeans(D^{-1/2}V^*, K)$;
end

Algorithm 1: K-way Constrained Spectral Clustering Algorithm

Algorithm 1 describes the CSC. The input required is an affinity matrix denoted A, a constraint matrix denoted Q with a corresponding adjustable belief parameter β, and the

number of clusters returned K. The matrix A represents a normalised pairwise similarity relation between data points with values ranging from 0 to 1, where 0 means *not similar* and 1 means *similar*. The constraint matrix Q is a normalised belief matrix with values ranging from -1 to 1, where each value is a soft measure of belief regarding whether two data points should be in the same cluster. The values -1, 0, and 1 should be interpreted as *Cannot-Link*, *no information*, and *Must-Link* respectively. The sum of all values in A is denoted *vol* and the diagonal matrix D is defined with diagonal values as the sum of each row in A, as can be seen in line 1. The normalised Laplacian matrix is defined on line 2 as $\bar{L}$ as well as the normalised constraint matrix denoted as $\bar{Q}$. The parameter β should be in the range of $(\lambda_{min}(\bar{Q}) \cdot vol, \lambda_{K-1}(\bar{Q}) \cdot vol)$, where $\lambda_{K-1}(\bar{Q})$ is the $(K-1)$-th largest eigenvalue of $\bar{Q}$. If β is in this range, then there exists at least $K-1$ eigenvectors associated with positive eigenvalues, and a solution for the constrained spectral clustering is guaranteed. The upper bound of this range is checked on line 3 to line 5 and an empty result is returned if β is outside. The generalised eigenvalue problem is then solved in line 7 and afterwards the non-positive eigenvalues and their corresponding eigenvectors is removed. The remaining eigenvectors are normalised as seen in line 8. On line 9, matrix V^* is then assigned the combination of $K-1$ eigenvectors that minimises the cut. The columns of V^* define a new space where the points are trivially clustered with K-means clustering. On line 10 K-means is run with the relaxed matrix, $D^{-1/2}V^*$, and K as parameters, that in turn provides a cluster assignment indicator $\mathbf{u}^*$.

4. Approach

The following section specifies the approach for constructing the affinity and constraint matrices. We address the issue of determining the number of clusters in an image, as standard spectral clustering requires this as an input parameter. A post-processing step which labels the detected clusters as either building clusters or non-building clusters and filters noise from the clustering is also presented.

4.1. The algorithm

Algorithm 2 gives an overview of our approach. The datasets defined as inputs of the algorithm have been described in Section 3.1. The construction of matrices A and Q is described in more detail in Sections 4.2 and 4.3 respectively. CSC have just been described in Section 3.2, elongated K-means are described in Section 4.4. Finally the post-processing will be described in Section 4.5.

Input : Aerial image, terrain-normalised object map, list of addresses
Output: Building clustered binary image
1 Construct the matrices A and Q
2 Run constrained spectral clustering with elongated K-means as the last step
3 Label each cluster with an address as a building cluster
4 Apply majority filter to the binary image

Algorithm 2: Building Detection

4.2. Affinity matrix construction

The process starts by organising the pixels of an image based on their colour (the three dimensional RGB space). Modern image segmentation techniques typically use more refined features for calculation the similarity between pixels, but for our example we restrict pixel similarity to the RGB space. For each pixel in the image, we only compute similarity with pixels that are no further away in the RGB space, than a certain threshold. Computing the similarity between each pair of pixels would have $O(n^2)$ complexity, where n is the number of pixels. We use a 3D RBF kernel (unnormalised spherical Gaussian) as the similarity measure. The affinity matrix A is defined as in Eq. (1).

$$A_{ij} = \begin{cases} s(p_i, p_j) & \text{if } s(p_i, p_j) \geq t_s \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

Note that the threshold t_s determines the sparsity of A, which is important for both memory usage and the computation time in spectral clustering. Dense affinity and constraint matrices easily become computationally problematic as n increases. A kd-tree [11] is used to efficiently find similar pixels in the image.

4.3. Constraint matrix construction

We construct two constraint matrices Q_{map} and Q_{point}. The matrix Q_{map} is based on the LIDAR data combined with the aerial image, whereas the matrix Q_{point} is based on the address point data combined with the LIDAR data. For both constraint matrices we will want to keep them sparse by creating only the most necessary constraints. We combine these two matrices into a final, combined constraint matrix Q defined as in Eq. (2).

$$Q_{ij} = \begin{cases} Q_{map,ij} & \text{if } Q_{point,ij} = 0 \\ Q_{point,ij} & \text{otherwise} \end{cases}. \tag{2}$$

Note that the construction of Q_{point} creates constraints based on addresses which we consider more trustworthy than the purely LIDAR based constraints. The constraints of Q_{point} will therefore always be selected over the constraints of Q_{map} when available. A discussion of how to combine constraints can be found in Section 6.1.

The construction of Q_{map} reuses a kd-tree indexing of the LIDAR map data, to efficiently add constraints between pixels that are very similar in colour and height. These constraints add necessary information about areas that may look similar but actually are at different heights, such as a building with a dark rooftop that casts a shadow on the ground besides it. We use a sigmoid function c defined as in Eq. (3) to create a constraint value based on the height difference between two pixels h_i and h_j on the height map.

$$c(h_i, h_j) = 2\left(\frac{1}{1 + e^{(c_s(|h_i - h_j| - c_o))}} - 0.5\right) \tag{3}$$

The parameter c_s controls the shape of the sigmoid and c_o is the offset of the function which determines what height difference should give the constraint value zero. We define Q_{map} as in Eq. (4).

$$Q_{map,ij} = \begin{cases} c(h_i, h_j) & \text{if } s(p_i, p_j) \geq t_s \\ 0 & \text{otherwise} \end{cases}. \tag{4}$$

In the construction of Q_{point}, we can map each registered address point to a specific pixel on the LIDAR height map. We do not consider addresses that does not appear to be on an object on the LIDAR height map. From each address point a breadth-first search is done by comparing the current pixel to every adjacent pixel. Doing this discovers neighbouring pixels that do not differ more than a certain threshold in height and the search continues from these adjacent pixels. Pixels that are grouped together are likely to be from the same object on the map. A Must-Link constraint is added between each of these pixels and every other believed pixel in the object. With this method we can also identify the pixels on the boundary of this object and add Cannot-Links between these and the pixels belonging to the object. Formally, we denote the j'th pixel with an address point as a_j. The height difference between two pixels h_i and h_j is $|h_i - h_j|$. We say h_i is *connected* to h_j if there exists a pixel path from h_i to h_j where, for each pixel h_k on the path $|h_k - h_j| \leq t_b$ and t_b is a building height threshold. We can define two sets of pixels M and C for every address point a_j. The set M contains all pixels connected to an a_j and the set C contains all pixels that are not connected to a_j but are adjacent to a pixel connected to an a_j.

We define Q_{point} using the sets of M and C as in Eq. (5).

$$Q_{point,ij} = \begin{cases} 1 & \text{if } p_i \in M \text{ and } p_j \in M \text{ or} \\ & p_i \in C \text{ and } p_j \in C \\ -1 & \text{if } p_i \in C \text{ and } p_j \in M \text{ or} \\ & p_i \in M \text{ and } p_j \in C \\ 0 & \text{otherwise} \end{cases}. \tag{5}$$

4.4. Determining the number of clusters

The presented CSC algorithm does, like USC, take the number of clusters K as input. Such an approach is not possible in our case as we want our algorithm to work with arbitrary aerial images where the number of cluster segments can vary significantly.

We will instead adopt a method for automatically determining the number of clusters, as described in [12]. The authors make the observation that similar data points in the cluster space defined by a set of eigenvectors, will cluster along radial lines. Using this behaviour of the clusters, the authors move on to present a modified K-means algorithm, called elongated K-means, where the distance along non-radial lines between points and cluster centres are penalised in the distance function.

4.5. Post-processing

The elongated K-means may return a large number of clusters, therefore, we will have to merge the clusters into two super-clusters; a building-cluster and a non-building-cluster. Since there is no direct indication of which clusters are building clusters and which are non-building clusters we must turn to our data for confirmation. Any cluster containing a pixel associated with an address are labelled as a building-cluster. All other clusters

are labelled as non-building-clusters. Note that the labelling method does not mean that buildings without addresses cannot be labelled as buildings, since they may be clustered with other buildings with addresses. This could either be because the buildings are physically connected or they are similar in colour and height. The result of the labelling is a binary image.

The clustering may have been noisy, and the binary image may thus also be noisy. In [13] they addressed the same problem with noise in clustering of satellite images. Their method for reducing the noise in binary images was to apply binary morphological operators as filters. We will adopt the same approach and apply a majority filter to the binary image as the last step. The majority filter will set a building pixel as non-building if m of the 8-connected neighbouring pixels are non-building pixels.

5. Evaluation

Our evaluation will focus on our method for creating constraints based on map and point data. We will run CSC with the two types of constraints individually and combined. We will also compare this performance to USC, which is equivalent to setting $\bar{Q} = I$. Performance will be presented as precision and recall over a confusion matrix created by comparing the results of our algorithm to ground truth clustered images.

Before evaluating performance, we will tune the parameters of our algorithm. For tuning and evaluation we will have two distinct datasets each of 10 aerial images. Each image will also have an accompanying height map, list of addresses on that image, and a ground truth clustered image. For practical convenience, the resolution of the images will be scaled to be equal to the resolution of the height map, such that there is a one-to-one correspondence between pixels in the two images.

5.1. Parameter tuning

Our algorithm contains eight parameters: The constraint belief β in the CSC, the elongation sharpness λ in elongated K-means, the parameters c_s and c_o for the sigmoid function, the threshold t_b in the creation of Q_{point}, the threshold t_s in the creation of A , the width of the Gaussian distribution σ in the 3D Gaussian, the m neighbouring pixels in the majority filter. For performance we want to keep the matrices sparse by setting t_s high, but we do not want to leave out much of the Gaussian distribution. Therefore we want to set σ and t_s such that we keep at least 95 % of the Gaussian distribution. In our case, we obtain this by querying on the kd-tree for pixels whose RGB values all are less than 60 in the range $(0,255)$. When we also satisfy the condition that we keep 95 % of the Gaussian distribution, then there exists only one solution for σ and t_s. Because we set σ and t_s this way, we do not tune them afterwards.

The authors of [5] suggests that β should be set in the range $(\lambda_{min}(\bar{Q}) \cdot vol, \lambda_{K-1}(\bar{Q}) \cdot vol)$. Since vol and $\bar{Q}$ are dependent on the image being clustered, we need an approach for tuning β that is independent of both vol and $\bar{Q}$. Our approach will be to introduce two other parameters i and p and set β as in Eq. (6).

$$\beta = \lambda_i(\bar{Q}) \cdot vol - p(\lambda_i(\bar{Q}) \cdot vol - \lambda_{i+1}(\bar{Q}) \cdot vol), \tag{6}$$

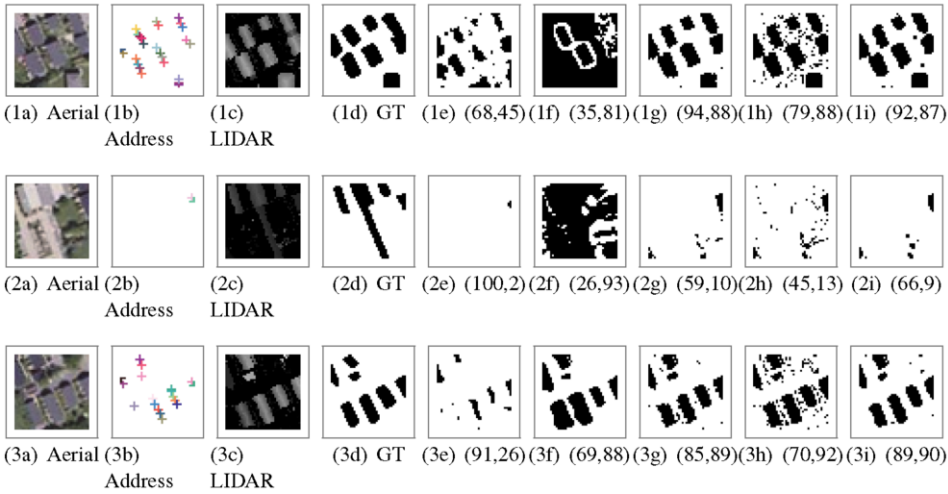

Figure 1. Test results of three images. Aerial (a). Address (b). LIDAR (c). Ground truth (GT) (d). USC (e). CSC with only Q_{point} (f). CSC with only Q_{map} (g). CSC with combined constraints Q unfiltered (h) and filtered (i). Results are represented by (Precision, Recall) in pct.

where $\lambda_i(\bar{Q})$ is the i-th largest eigenvalue of $\bar{Q}$ and p is in the range $0.1, 0.2 \ldots, 1$. The value of p will determine how close β is to the range bounds. Tuning i is equivalent to finding the optimal lower bound number of eigenvectors, that will be used in the elongated K-means algorithm.

It is not practical to test all combinations of the remaining parameter values, so the parameters will instead be tuned in turn until a local maximum in accuracy is found. We will tune the parameters for the four evaluation cases: USC, CSC with only the constraints Q_{point}, CSC with only the constraints Q_{map}, and CSC with the combined constraints Q.

5.2. Evaluation results

Figure 1 shows the results of the evaluation on 3 aerial images. The result sets are labelled from 1 to 3. The images labelled from a to d are aerial image, address locations, height map and ground truth clustering respectively. Images labelled e are the result of unconstrained clustering with majority filter applied. Images labelled f and g are the result from CSC with only constraint from Q_{point} and Q_{map} respectively, both with majority filter applied. Images labelled h and i are the results from incorporating both constraint types and shows the building detection before and after the majority filter has been applied.

As the USC does not use any of the constraints built, then of course it is entirely dependent on its similarity function as the basis of its final clustering. In these clusterings it is clear, that the USC is not able to cluster buildings correctly on its own. The clustering quality of only using the constraints built from address point data (Q_{point}) has shown to achieve an average precision of 26 % and a recall of 76 %. The clustering using when using only Q_{point} have shown difficulties in separating some houses from the terrain resulting in labelling most of the image as a building. By visually interpreting the

clustering quality of only using Q_{point} this bad performance can be verified. Tests showed that the clustering quality achieved using only Q_{map} is closest to the one achieved when using the combined constraints with an average precision of 85 % and recall of 63 %. However in some cases, parts of the terrain were included in building clusters. Many of these incorrectly clustered pixels were then corrected when combined with Q_{point}. This is especially visible in image set 2.

The majority filter effectively removes most noise from the unfiltered clustering. Due to the nature of the filtering method, it increases precision by removing the noise but it also smooths out structures in the clustering that in some cases result in decreased recall.

Image 2 have one or more buildings that have not been labelled as buildings in the clustering. This is in most cases a result of a building cluster lacking a corresponding address point, and the clusters is therefore labelled as non-building. This problem will most likely be mitigated by using larger images, where these buildings would be clustered with other buildings of similar colour with address points.

Combining both constraints shows the best results with an average precision of 88 % and recall of 62 %. The results are in the vicinity of methods for building detection, even though it was not our main purpose to compete with state-of-the-art building detection algorithms. Examples of methods for building detection include a method from [10] that has a precision of 90 % and recall of 88 %. Another paper [1] has shown a precision of 89 % and recall of 93 %.

6. Discussion

Further improvements of the method for combining constraints will be discussed in Section 6.1. The use of multiple constraint types and possible other applications for this combination is discussed in Section 6.2.

6.1. Constraint combination methods

We suggest two general methods for combining constraints, created from several sources, into the the combined constraint matrix Q. The first method is to define a hierarchical ordering of the constraint matrices and let non-zero valued constraints overrule any lower ordered constraint values. We utilised this method for our application of building detection. The second method is the linear combination of constraints based on their relative importance. Both of these techniques are trivial to extend to more than two sources of constraints.

6.2. Use of constraint types

We have shown how to use and combine constraints which are generated from the two data types; maps and points. More generally these two constraint types can be used in such a way that they are applicable in other image clustering applications.

Having data in the form of a map provides each pixel with additional data such that a pixel does not only contain a colour value. This requires a relation that maps pixels from the image to values on the map. In our application, this relation was a bijection from

the aerial image to the height map. By converting the LIDAR height data to a greyscale image we showed that the map data itself can be images.

Data points had two uses in our application. The first is for semi-supervised labelling of clusters. The second use is that they are indicators of objects on a map. This use requires some method for identifying the object boundaries, which in our application was any neighbouring pixel differing no more than a threshold in height.

7. Conclusion

We proposed a new approach for combining constraints built from a combination of multiple sources of public data. The datasets were data represented as both single data points as well as map data. The datasets were processed and combined based on their relative trustworthiness. The main purpose of this paper was not to obtain the highest accuracy of the building detection/image segmentation, but instead to present an approach for combining multiple datasets into one set of constraints to be used in CSC. Furthermore this paper presented a real application in which this approach can be utilised.

The evaluation showed that the two constraint types supplemented each other as the clustering quality was shown to be best when the constraints were combined. The average quality achieved was a precision of 88% and a recall of 62%, which places our method in the vicinity of other building detections methods, even though it was not the main purpose of this paper.

References

[1] M. Awrangjeb, M. Ravanbakhsh and C. S. Fraser, Automatic building detection using lidar data and multispectral imagery. *Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference on*, IEEE, (2010), 45–51.

[2] Kortforsyningen, `http://www.kortforsyningen.dk/`, Accessed: 12-4-2013.

[3] Bygnings- og boligregisteret, `http://www.bbr.dk/`, Accessed: 15-4-2013.

[4] Offentlligt data - digitaliser.dk, `http://digitaliser.dk/group/237756`, Accessed: 19-3-2013.

[5] X. Wang, B. Qian and I. Davidson, On Constrained Spectral Clustering and Its Applications, *CoRR*, (2012).

[6] U. Luxburg, A tutorial on spectral clustering, *Statistics and Computing*, (2007), 395–416.

[7] S. D. Kamvar, D. Klein and C. D. Manning, Spectral learning, *IJCAI*, (2003), 561–566.

[8] Q. Xu, and M. Desjardins, Constrained Spectral Clustering under a Local Proximity Structure Assumption, *Proceedings of the 18th International Conference of the Florida Artificial Intelligence Research Society*, (2005).

[9] D. B. Tijl, J. Suykens and D. M. Bart, Learning from General Label Constraints, *Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshops*, (2004), 671-679.

[10] G. Sohn and I. Dowman, Data fusion of high-resolution satellite imagery and LiDAR data for automatic building extraction, *ISPRS Journal of Photogrammetry and Remote Sensing*, (2007), 43–63.

[11] Moore, An introductory tutorial on kd-trees, *Extract from Andrew Moore's PhD Thesis: Effcient Memory based Learning for Robot Control*, (1991).

[12] G. Sanguinetti, J. Laidler and N. D. Lawrence, Automatic determination of the number of clusters using spectral algorithms, *IEEE Machine Learning for Signal Processing*, (2005), 28–30.

[13] Y. Li, J. Li and M. A. Chapman, Automatically Extracting Manmade Objects from Pan-Sharpened High-Resolution Satellite Imagery Using a Fuzzy Segmentation Method, *Geo-information for Disaster Management*, (2005), 641-653.